SECURITE DES SYSTEMES INFORMATIQUES

COMPTE RENDU TP6

FATOUMBI AFDOL

Arnold NGNOUBA NDIE TCHEGOU

Question1 : Quel est la période du PRNG

Pour déterminer la période du générateur pseudo-aléatoire (PRNG), nous avons écrit une fonction qui parcourt le fichier **prng.out** et s'arrête dès qu'elle rencontre le même élément pour la deuxième fois.

Question2 : Retrouvez la clé secrète d'alice et Bob

Pour retrouver les clés secrètes, nous avons examiné la fonction **leaking_square_and_mult**. Dans cette fonction, nous avons noté que chaque fois que le bit de l'exposant est égal à 1, nous écrivons '10' dans le fichier **sqr_and_mult.out**, et '0' lorsque le bit de l'exposant est égal à 0. Par conséquent, nous avons créé deux fonctions : une fonction qui supprime les zéros qui apparaissent juste avant les '1', et une autre fonction qui inverse la chaîne obtenue pour reconstituer la clé secrète dans chaque cas. Les résultats obtenus sont : **OK!!** Pour alice et **GG!!** Pour bob.

Question3 : Retrouvez le challenge

Nous avons identifié une vulnérabilité dans la fonction responsable de la création de la clé partagée et du chiffrement du challenge. Cette fonction enregistre dans le fichier **"transcript.out"** la valeur des premiers 4 octets de la clé de chiffrement. Étant donné que le fichier **"prng.out"** est public, nous sommes en mesure de récupérer les trois autres blocs de la clé de chiffrement. Par la suite, nous pouvons procéder à l'élaboration d'une fonction de déchiffrement pour récupérer le challenge. Le résultat du challenge est : **Correct chall!!!**

Question4 cas 1: Eve doit faire déchiffrer « send 100$ to Eve à Bob »

Le protocole implémenté ne permet pas d'authentifier les parties prenant part à une communication. Ayant observé cela, nous avons procédé comme suit :

- Eve génère sa clé publique et la diffuse dans le fichier alice_pub_key.
- Eve initialise la communication avec Bob en lui envoyant le challenge chiffré, utilisant sa propre clé secrète et la clé publique de Bob qu'elle récupère dans le fichier bob_pub_key.
- Bob déchiffre ensuite le challenge en utilisant la clé publique d'Eve qu'il récupère dans le fichier alice_pub_key et sa propre clé secrète.
- Eve chiffre le message en utilisant le challenge comme vecteur d'initialisation (IV) pour le chiffrement AES-CBC, puis elle l'envoie à Bob. Ce dernier le déchiffre à son tour en utilisant le même challenge.

Question4 cas 2: Eve intercepte le message et le modifie

Nous n'avons pas implémenté ce cas en langage c, mais avec la même observation que précédemment, nous avons pensé ceci :

- Eve initie une communication avec Alice et Bob en se faisant passer pour chacun d'entre eux dans chaque cas. Pour ce faire, Eve doit récupérer les clés publiques d'Alice et de Bob dans les fichiers pub_key respectifs, puis elle écrit sa propre clé publique à la place.
- Eve intercepte ensuite les messages, car les messages envoyés par Alice et Bob lui sont destinés. Elle peut ainsi remplacer un message chiffré par son propre message chiffré et le renvoyer à chacun d'entre eux.